逆向分析实验2

[1. 实验内容 1](#_Toc28113)

[2. 实验准备 1](#_Toc20435)

[3. 实验过程 1](#_Toc31553)

[3.1. 逆向Android程序 1](#_Toc4295)

[3.2. 逆向so程序 1](#_Toc9269)

[3.3. “flag”的分析与跟踪 1](#_Toc23451)

[4. 扩展 2](#_Toc22343)

[5. 实验成绩评定 2](#_Toc23005)

# 实验内容

结合Android程序实例“AndroidCrackme.apk”，完成以下工作：

1. 分析Android程序执行的过程，及核心代码的地址和范围；
2. 程序中关键信息混淆的方法，并还原相关内容；
3. 程序中反调试方法；
4. 还原程序中被加密的代码，结合还原后的代码重新打包生成程序，并确保程序能够正常运行；
5. 在上述工作基础上，获取实例代码中的“flag”，并进行验证。

# 实验准备

1. Android手机；
2. AndroidKiller软件；
3. IDA Pro 7.0软件。

# 实验过程

## 逆向Java代码

反编译实例程序，确定关键函数位置。

## 逆向so程序

1. 借助于IDA工具，逆向实例程序，根据Android程序执行过程，定位核心代码；
2. 分析程序在初始化过程中，如何完成对关键信息的混淆，根据分析的结果，还原这些关键信息；
3. 分析程序中反调试的方法；
4. 借助于指令修改工具，去除反调试代码；
5. 通过静态分析方法，对核心代码进行还原。

## “flag”的分析与跟踪

1. 根据反编译的伪C代码，跟踪相关信息；
2. 以“flag”可能涉及到的变量为线索，跟踪“flag”相关的信息；
3. 尝试恢复“flag”。

# 扩展

在去除反调试代码的基础上，通过查阅资料，学习动态调试的方法，并尝试采用动态分析的方法获取“flag”。

# 实验成绩评定

实验成绩的评定主要依据包括两个方面：

1. 文档的规范性和细节的描述。规范性体现的是态度；细节则是分析工作的深入程度。
2. 分析的结果，是否达到了实验的一些基本要求。